Remarks

The Office Action dated February 12, 2003 has been noted and its contents carefully studied. In light of the foregoing amendments, reconsideration of the rejection under 35 U.S.C. 102(b), (e), and 103(a) is courteously requested.

In order to facilitate the Examiner's reconsideration, the following discussion of the invention as reflected in the amended claims is presented hereinafter. Support for the amendments to the claims is found in paragraphs 0022, 0023 and 0024.

In one aspect, the invention relates to a method of searching a file access system for a requested file. As currently recited in the corresponding amended claims, the method includes establishing a field in a directory i-node memory structure for files corresponding to a directory cache hash table. The field contains a pointer to the directory cache hash table, and the directory cache hash table is searched for a requested file by hashing the file i-node to a specific bucket which contains a list of files that may correspond to the requested file i-node. If the bucket contains a matching file name then a pointer is established to where the name of the requested file is stored to thereby reduce extensive sequential searching normally required.

In another aspect, the invention relates to a method of accessing files in a file access system which includes the aforementioned establishing of a field in a directory i-node memory structure. In this specific aspect, the directory is converted from a storage device representation to a faster representation. The faster representation represents a layout of the directory with an array of hash buckets which point to a list of files which

may correspond to a specific i-node, and the method proceeds as previously discussed. These features are all found in independent claims 1, 5, and 10 as presently amended.

In an alternative aspect as reflected by independent claim 14, the invention relates to a computer server system. The computer server system includes directory layouts arranged in a faster representation than that of a storage device layout. The faster representation includes an array of hash buckets which point to a list of files which may correspond to a specific i-node. The faster representation further includes a pointer from a directory i-node memory structure to an associated hash table. In yet another aspect, as reflected in claim 15, the invention relates to a network storage system, which as amended includes the aforementioned features discussed with reference to claim 14.

Having thus generally discussed the invention, it will become readily apparent to the Examiner that the invention is not anticipated by or obvious from the cited references, based on the following discussion of the references presented herein for the Examiner's kind consideration.

U.S. Patent 6,192,408 to Vahalia, et al

U.S. Patent 6,192,408 to Vahalia, et al (hereinafter Vahalia) discloses a method of servicing data access requests from clients in a data network. The data network has a first set of data processors receiving client requests for access to a plurality of read-write file systems. A second set of data processors manages locks on the read-write file systems, and a respective one of the data processors in the second set of data processors is presently assigned to manage locks on each of the read-write file systems (column 1,

lines 35-41). With respect to the discussion of the cached disk storage subsystem storing the file systems, the network file server has a cached disk storage subsystem storing the file systems and a multiplicity of data mover computers linking the cached disk storage subsystem to the network for servicing the request by transferring data between the clients and the cached disk storage subsystem (column 2, lines 23-27).

In this regard, it is noted that the cache is used to store file data. There is nothing in Vahalia which teaches or suggests establishing a field in a directory i-node memory structure for files corresponding to a directory cache hash table and then searching the directory cache hash table by hashing the file i-node to a specific bucket which contains a list of files that may correspond to the requested file i-node. If found, the bucket then merely contains pointer information to where the name of the requested file is stored so that the file can be directly accessed without searching through directories in a conventional manner. These are features found in all of the independent claims and not taught or suggested by Vahalia alone or in combination with any of the other references cited.

It is important to appreciate that what Vahalia teaches is shortening searches for file data by caching selected files in a disk cache, but there is no teaching or suggestion therein of a combination with a directory i-node memory structure establishing a directory cache hash table through which conventional searching for files through directories can be avoided through what is essentially a "shortcut" mechanism.

With respect to claims 2 and 3, they recite further features of the invention which if, in one case, when the file name in the directory cache hash table is not found, conventionally searching the file structures.

Since it would be generally physically difficult and would require a very large directory cache hash table to include all of the directories' information in the file access system, the directories to be identified in the directory cache hash table are selected using at least one of the number of files in a directory, and the frequency of use, which are features which are clearly not taught or suggested by Vahalia which fails to recognize or implement a system employing the i-node memory structure with pointers to a directory cache hash table (as contrasted with conventional cache for file data) in a manner set forth in claims 1-3.

Thus, for the foregoing reasons it is respectfully urged that the invention as recited in claims 1-3 is not anticipated or obvious from Vahalia and/or the other references which are discussed hereinafter.

U.S. Patent 5,666,532 to Saks, et al

U.S. Patent 5,666,532 to Saks, et al (hereinafter Saks) has been cited to reject claims 4-5, and 14-15 under 35 U.S.C. 102(b) as anticipated. It is respectfully urged from the following discussion that the claims as currently amended are not anticipated by Saks for the reasons set forth hereinafter.

More specifically, Saks teaches a computer system having one or more types of file subsystems for controlling transfer of files between primary storage and secondary

storage. A subset of writes to secondary storage is performed using a delayed ordered right (DOW) subsystem which makes it possible for any file system to control the order in which modifications are propagated to disk. The subsystem consists of two parts. A first part is a specification interface which a file system implementation can use to indicate sequential ordering between a modification and some other modification of file system structural data. This results implicitly in the construction of an order store in primary storage that records the ordering interdependence among different buffers affected by the modifications (column 2, lines 54-67).

In the file system, the internal representation of a file is given by an i-node which contains a description of the device layout of the file data and other information. The system recognizes that every file or directory has one i-node but may have several names, all of which map into the same i-node. When a process creates or opens a file by name, the system parses each component of the file name, checking that the process has the necessary privileges for searching the intermediate directories and then retrieves the i-node for the file. In the case of opening or creating a new file, the system assigns to the file an unused i-node and i-nodes are stored in the file system, and needed i-nodes are read into an i-node table that is maintained in primary storage for access (column 8, lines 20-28).

The blocks and storage that contain file data are reachable from the i-node for the file by one of three means. The i-node itself contains the logical addresses for several initial data blocks. In addition, the i-node contains a few block addresses for "indirect" and "double indirect" blocks for large files (column 8, lines 50-54).

When a process is to access data from a file, the data is brought into main storage where the process can examine it, alter it, and request that the data can be saved in the file system again. In all cases, as file sizes and file and directory populations in a file system change, auxiliary data that describes the file system organization is brought into memory, examined and altered, and written to secondary storage. The file system attempts to minimize the frequency of disk access by keeping file data and file system structural information in page cache or buffer cache.

In this regard, it is noted that there is nothing in Saks which teaches or suggests, as reflected in independent claims 4, 5, 10, 14 and 15, the concept of establishing a field in a directory of i-node memory structures for files corresponding to a directory cache hash table, and with the field containing a pointer to the directory cache hash table, and further having a representation of a layout of the directory within an array of hash buckets which point to a list of fields which may correspond to a specific i-node so that a requested file can be searched by hashing the file i-node to a specific bucket which contains the list of files that may correspond to the requested file i-node, from which the exact location in memory the requested file and corresponding data can be found.

Claims 6-9 recite various additional features of how selected directories are hashed into a hash table format, notably, as previously discussed with respect to claim 3, claims 7-9 recite alternative means of selecting the directories.

Likewise, claims 11-13 recite selection criteria in a manner similar to claims 7-9.

These features are clearly not taught or suggested by Saks alone, or in combination with other references discussed hereinafter. Thus, for the foregoing reasons

it is respectfully urged that claims 4-5 and 14-15, as well as claims 6-9 to which Saks has been applied, are clearly patentable over the teachings of Saks.

U.S. Patent 5,778,420 to Ish, et al

U.S. Patent 5,778,430 to Ish, et al (hereinafter Ish) merely discloses a computer disk cache management method and apparatus which employs a least-recently-used with aging method to determine a best candidate for replacement as a result of a cache miss. A hash table points to a block of data contained within cache. This is nothing more than conventional caching of data with a specific mechanism for replacing data which is least frequently accessed by new data which may be accessed more commonly.

As noted previously with respect to Saks as applied in combination with Ish to reject claims 6-13, this has nothing to do with the claimed invention and adds nothing to the teachings of Saks which also employs cache primarily for the purpose of storing data.

As already noted, the present invention implements an i-node structure with pointers to a hash table containing buckets which may or may not contain information pointing to the exact location on disk for directories, files and data sought to be accessed. This is not taught or suggested by any of the references.

The remaining references have been reviewed and are not considered any more pertinent to the invention than the applied references discussed previously.

For the foregoing reasons, it is respectfully urged that the claims are now in condition for allowance. Nonetheless, should the Examiner have any comments,

questions or suggestions of a nature necessary to expedite prosecution of the application, he is courteously requested to telephone the undersigned at the number listed below.

Respectfully submitted,

A. José Cortina, Reg. No. 29,733
One of the Attorneys for the Applicant
Daniels Daniels & Verdonik, P.A.
Post Office Drawer 12218
Research Triangle Park, NC 27709
Telephone (919) 544 5444
Facsimile (919) 544 5920
Email JCortina@d2vlaw.com

Customer No. 26108

Dated June 11, 2003

F:\CL\130. -003\Prosecution\Response to OA.DOC